

АВТОМАТИЗОВАНА СИСТЕМА ОБЛІКУ ВІДВІДУВАНЬ ЗАНЯТЬ І ГЕНЕРАЦІЇ ЗВІТІВ

Стьопін В.І., студент кафедри СТ, e-mail: vladyslav.stopin@nure.ua

Горбатенко Б.В., студент кафедри СТ,

e-mail: bohdan.horbatenko@nure.ua

Науковий керівник: к.т.н., проф. Іванов В.Г.

Харківський національний університет радіоелектроніки

A web-based system for registration student visits is proposed. Nowadays, there are a lot of ready solutions that implemented in working flow of organizations and educational establishments. But all of them require additional resources, such as time or equipment. Developed system allows avoiding large expenditures, because of using smartphones and Internet technologies. Main business processes of the system are visits recording and statistics gaining. This approach of registration minimizes time spendings for users accounting, creating reports and analyzing visits statistics.

На багатьох підприємствах, в усіх школах та університетах є система контролю за відвідуванням певних заходів. Таким чином керівні органи можуть контролювати процеси, які проходять всередині їх структури. З розвитком сучасних технологій це стало робити простіше завдяки електродним таблицям, Інтернету та безконтактними технологіями. Але навіть сьогодні, в учбових закладах, ще витрачається значний час занять на перевірку наявності студентів. Для вирішення цієї проблеми була розроблена система, яка дозволяє студентам самостійно відмітити себе, як присутнім на занятті. Такого роду система є першою в сфері освіти. Вона призначена для спрощення роботи викладача та університету в цілому завдяки можливості ведення електронних журналів та світових документів. Це зводить до мінімуму затрати на вирішення організаційних питань за рахунок навчального часу, а також створення звітів.

Системи, які реалізують такий принцип автоматизації вже існують, але мають значні недоліки. Деякі з них застарілі для сучасного світу, мають проблеми з безпекою персональних даних, складні у використанні або потребують спеціальних знань, та інші. Розроблена система закриває ці питання, але залишає одну важливу

умову – для роботи з системою треба мати смартфон та доступ до Інтернету, щоб виконати поставлену задачу.

Ціль системи – знизити затрати часу на відмічання присутності студентів під час занять або виключити їх взагалі, ведення повного обліку по відвідуванню [1]. Функціонал системи пропонує декілька способів досягання цілі. Перший – завдяки скануванню студентами спеціального коду, який буде у викладача. Другий – повна автоматизація, тобто студенту достатньо знаходитися на занятті з включеним смартфоном, який підключений до Інтернету. За основу були взяті вже існуючі сервіси та системи, які гарно себе зарекомендували за останній час. Мета системи – зробити облік по відвідуванням швидко доступною, зрозумілою для кожного співробітника університету та студентів, які можуть передивлятися свої успіхи [2].

Функціонування сервісу представляє собою реалізацію бізнес-процесу, який вирішує основне питання – відмітити в електронному списку студента. Представимо такий процес за допомогою діаграми IDEF0 для першого способу, коли на вхід системи отримуємо код викладача (рис. 1). Тоді, на виході, система повинна сказати, студент був присутній, чи може код викладача не вірний, або інші випадки наприклад, коли студент вирішив прийти не на свою пару. Регулюється цей процес політикою конфіденційності, так як в системі зберігаються персональні дані користувачів. Це означає, що канали передачі даних повинні бути зашифровані, а доступ о даних – обмежено правами та політиками системи. В системі є три основні ролі: студент, викладач і сама система, як сукупність внутрішніх та зовнішніх функцій.



Рисунок 1. IDEF0 діаграма процесу перевірки коду викладача

Після сканування коду, студент переходить по зашифрованому посиланню, яке веде у систему. Для того, щоб активізувати код – потрібно пройти авторизацію. Для студентів, які є в системі – достатньо ввести свій логін та пароль. Реєстрація проходить з підтвердженням, тобто не кожний бажаючий може стати користувачем, а лише ті, чиї акаунти підтвердить адміністратор. Після авторизації – система обробляє повідомлення з коду та порівнює його з кодом, який записаний в базі даних. Якщо вони збігаються, це значить, що студент прийшов на своє заняття та був присутній. Якщо код не співпадає, буде видана помилка. Помилки можуть виникати також за умов: студент прийшов на пару не за розкладом, код застарілий, системна помилка. Останній тип помилок не можна повністю виключити, так як система знаходиться на серверах.

Іншим способом відмічання є використання геопозиції студента, позначена на (рис. 2). Для цього потрібно лише, щоб користувач системи дозволив своєму пристрою відстежувати його положення (натиснути «Погоджуюся» у вікні ОС). Тоді, система може автоматично відстежити координати студента у заданий час, коли повинні бути заняття в університеті. Погрішність такого метода складає 20 метрів. Для забезпечення достовірності особи студента також збирається і інша статистика, яка дозволить дізнатися, чи намагався студент обманути систему.

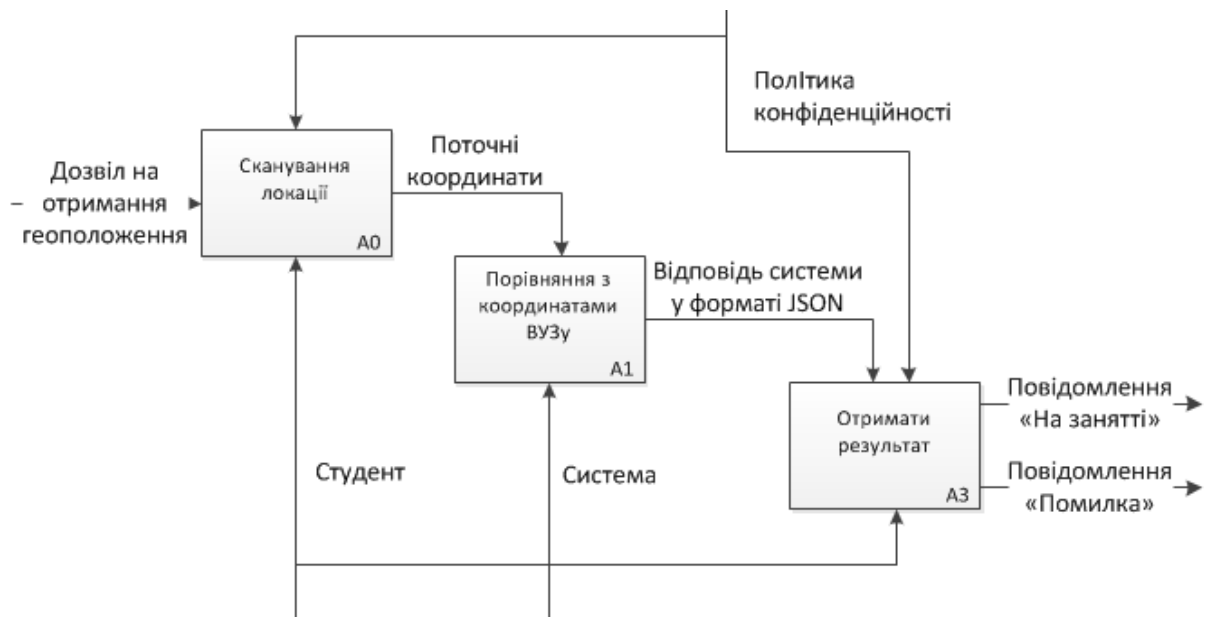


Рисунок 2. IDEF0 діаграма для процесу перевірки студентів на заняттях за їх геопозицією через телефон

Для початку роботи з системою потрібно бути зареєстрованим. Макет особистого кабінету показано на Рисунку 3. Процес реєстрації студентів можна зробити автоматичним за рахунок корпоративної пошти. При створенні аккаунту, його можливо одразу підключити до сервісу відмічання (через Google+ технологію). Викладачів треба реєструвати окремо, тому що для них буде генеруватись окремий код на кожний предмет за розкладом. Взагалі, процес реєстрації не потребує багато часу, навіть виключаючи можливості автоматизації цього процесу. Також існує функція реєстрації заявок на співробітництво з іншими структурами, де потрібна така система. Окрема реєстрація необхідна для контакту з адміністрацією, створення робочих аккаунтів для співробітників та менеджерів, тощо. Зліва від форми реєстрації знаходяться корисні посилання на інформацію щодо політики конфіденційності, часті питання та форма зворотного зв'язку з адміністрацією сайту [3].

Особисті кабінети студентів та викладачів відрізняються за структурою. У студентів можна побачити таблиці розкладу, їх статистику відвідування та пропусків занять, предмети та викладачів. У викладачів є списки студентів, таблиці, де зазначені відвідування, статистика по обліку студентів. Ця статистика включає: тип мобільного пристрою, браузер, операційну систему,

геолокацію, час реєстрації, браузер та інші характеристики, які можуть точно сказати, який саме студент виконував процедуру обліку. Це зроблено для безпеки та правильності даних, які поступають до бази.



Рисунок 3. Макет особистого кабінету

Базою даних для системи була обрана MySQL за її властивості надійності, гнучкості та можливості росту [4]. Основні таблиці, які були створені для функціонування – це таблиці користувачів (студенти, викладачі, адміністратори), заняття (електронний розклад), динамічні посилання та коди, а також окремі таблиці для різноманітної статистики, описаної вище. Щоб підвищити безпеку таблиць – їх ім'я повинно починатися з префіксу. Уся передача даних між таблицями та сервером буде зашифрована, а пакети будуть відправлятися у кросплатформерному форматі json. Для підвищення швидкості роботи – поля у таблицях оптимізовані під данні, що зберігаються, не тільки по формату, а й по величинам. Це дозволяє знизити навантаження на базу даних за умови великої кількості записів. Реляційна модель вже не так вражає, багато команд ІТ-бізнесу відмовляються від настільки складних рішень. Але стиль розробленої системи дозволяє використовувати нормалізацію бази даних, тому при невеликій кількості записів, таке рішення працювало повільніше, ніж не реляційні аналоги.

Основні характеристики системи – це надійність, швидкість роботи та цілісність. Для забезпечення таких умов був

використаний язык програмування PHP, який також дозволяє легко взаємодіяти с базами даних. Для обмеження виконання всього коду на сервері та розподілити навантаження між клієнтами – для виконання деякого коду використовується JavaScript бібліотеки (jQuery, Modernizr, LightBox, Lazy) та інші скрипти. Увесь проект розгалужений відповідно до моделі MVC. Окремі дані для доступу різних сервісів виділені в окремі конфігураційні файли, що збільшує надійність системи. Верстка сторінок відбувалась шаблонами, які обробляються сервером та наповнюються необхідними даними динамічно, в залежності від сторінки, яку завантажує користувач [5].

Таким чином, система автоматизації процесу обліку дозволяє значно знизити витрати часу на перевірку студентів, що знаходяться на занятті, а також вести електронну статистику, журнали та облікові документи. Завдяки інформації у електронному вигляді, існує можливість створення документів однією кнопкою. Переваги цієї системи: не потребує додаткових затрат на купівлю спеціального обладнання, інтуїтивно простий у використанні, не потребує спеціальних знань, здатний створювати облікові документи при необхідності. Із недоліків: працює тільки при умові підключення до мережі Інтернет, у студента повинен бути смартфон, для створення нових облікових записів потрібна ручна праця адміністратора ресурсу.

Використання обох способів обліку одночасно можливе, це призведе до збільшення точності інформації. Але також можуть виникнути помилки при записі даних до бази, протиріччя на стороні користувача або звичайна системна помилка. Рекомендується обрати один із зазначених вище способів. Зараз на практиці використовується спосіб з кодами викладача. Для розгортання такої системи в іншому університеті потрібно виконати наступні умови: мати приклади електронних документів органів управління університету, електронний розклад з прив'язкою до навчальних груп та викладачів, доступний Інтернет канал або мобільний зв'язок на навчальній території.

Систему можна модернізувати завдяки додатковим сервісам у панелі адміністратора, які поліпшать процес навчання. Так, для викладачів, існує можливість використовувати голосування, коли кожен студент в своєму смартфоні обирає відповідь, а на екрані викладача малюється графік відповідей. Або сервіс для обробки

персональних даних, наприклад для відкритих лекцій, в якому є можливість цифрової обробки паперових документів. Це збільшить навантаження на систему і зробить інтерфейс користувача складніше, що негативно вплине на нових користувачів, але значно підвищить сферу використання системи в навчальному та організаційному процесах.

Література.

1. Береза А.М. Основи створення інформаційних систем [текст] / А.М. Береза – К.:Лань, 2001. – 203 с.
2. Купер А., Інтерфейс. Основи проектування взаємодії. 4-е вид. [текст] / А. Купер, Р. Рейман, Д. Кронин, К. Носсед – СПб.: Пітер, 2017. – 720 с.
3. Сміт Д.М., Елементарні шаблони проектування [текст] / Д.М. Сміт, – М.: Вільямс, 2013. – 304 с.
4. Белл Ч., Забезпечення високої доступності систем на основі MySQL [текст] / Б.Чарльз, М. Кіндал – СПб.: БХВ-Петербург, 2011. – 624 с.
5. Ніксон Р. Створюємо динамічні веб-сайти за допомогою PHP, MySQL і JavaScript [текст] / Р. Ніксон, - СПб.: Ексмо, 2011. – 688 с.